IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s):	L-J. Zhang et al.
Docket No.:	SOM920030006US1
Serial No.:	10/665,699
Filing Date:	September 19, 2003
Group:	2454
Examiner:	Jeong S. Park
Title:	Methods and Apparatus for Information Hyperchain Management for On-Demand Business Collaboration

DECLARATION UNDER 37 C.F.R. §1.131

We, the undersigned, hereby declare and state as follows:

1. We are named joint inventors of the invention that is the subject of the above-referenced U.S. patent application. We have assigned our respective interests in the patent application to International Business Machines Corporation ("IBM").

2. We conceived an invention falling within the scope of the claims in the present application at least as early as July 30, 2003. We prepared a document, referred to as a "written embodiment" of IBM Disclosure SOM8-2003-0001, and including drawings, which was submitted to an IBM Patent Professional at least as early as July 30, 2003. A redacted copy of the written embodiment, including drawings, is attached hereto as Exhibit 1.

3. On or before July 30, 2003, an IBM Patent Professional, David M. Shofi, sent a letter to an IBM Outside Counsel, Ryan, Mason & Lewis, LLP, with instructions to prepare and file a U.S. patent application no later than September 30, 2003, based on the documents heretofore identified as Exhibit 1, which were enclosed with the letter. A redacted copy of the letter is attached hereto as Exhibit 2. Exhibit 2 contains an acknowledgement by Teresa M. Hamlin, Office Manager of Ryan, Mason & Lewis, LLP, that said letter was received on or about July 31, 2003.

4. Between July 31, 2003 and September 19, 2003, William E. Lewis of Ryan, Mason & Lewis, LLP, worked diligently on the preparation of the above-referenced patent application and communicated with us regularly to obtain further information and to keep us apprised of his progress.

5. On or about September 11, 2003, Mr. Lewis sent us a first draft of the above-referenced patent application. A redacted printout of an e-mail message dated September 11, 2003, from Mr. Lewis to inventors Tian Chao, Liang-Jie Zhang and John Sayah, to which this first draft was attached, is attached hereto as Exhibit 3.

6. Between September 11, 2003 and September 18, 2003, William E. Lewis of Ryan, Mason & Lewis, LLP, worked diligently to obtain comments from us responsive to the first draft and to prepare a revised draft in accordance with our comments. During this period, Mr. Lewis communicated with us regularly to obtain further information and to keep us apprised of his progress, as evidenced by, for example, the e-mail messages attached hereto as Exhibit 4.

7. Exhibit 4 includes a redacted printout of an e-mail message dated September 18, 2003, from Mr. Lewis to inventors Liang-Jie Zhang, Tian Chao and John Sayah, to which this revised draft was attached.

8. During the period from September 18, 2003, to September 19, 2003, Mr. Lewis worked diligently to obtain necessary approval from us and from Mr. Shofi.

9. The invention was constructively reduced to practice by filing the above-referenced patent application on September 19, 2003 with the U.S. Patent and Trademark Office.

10. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true.

11. We understand that willful false statements and the like so made are punishable by fine or imprisonment, or both, under §1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Date: 4/3/2009 ____________________
Liang-Jie Zhang

Date: __________ ____________________
John Youssef Sayah

Date: __________ ____________________
Tian-Jy Chao

Date: __________ ____________________
Ying Nan Zuo

Date: __________ ____________________
Shun Xiang Yang

Date: __________ ____________________
Jing Min Xu

Date: __________ ____________________
Haiyan Wang, signing on behalf of
deceased inventor Yiming Ye

10. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true.

11. We understand that willful false statements and the like so made are punishable by fine or imprisonment, or both, under §1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Date: ____________ ______________________________
Liang-Jie Zhang

Date: 4/2/2009 ______________________________
John Youssef Sayah

Date: ____________ ______________________________
Tian-Jy Chao

Date: ____________ ______________________________
Ying Nan Zuo

Date: ____________ ______________________________
Shun Xiang Yang

Date: ____________ ______________________________
Jing Min Xu

Date: ____________ ______________________________
Haiyan Wang, signing on behalf of
deceased inventor Yiming Ye

10. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true.

11. We understand that willful false statements and the like so made are punishable by fine or imprisonment, or both, under §1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Date: ____________ ______________________________
Liang-Jie Zhang

Date: ____________ ______________________________
John Youssef Sayah

Date: March 27, 2009 Tian-Jy Chao
Tian-Jy Chao

Date: ____________ ______________________________
Ying Nan Zuo

Date: ____________ ______________________________
Shun Xiang Yang

Date: ____________ ______________________________
Jing Min Xu

Date: ____________ ______________________________
Haiyan Wang, signing on behalf of
deceased inventor Yiming Ye

10. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true.

11. We understand that willful false statements and the like so made are punishable by fine or imprisonment, or both, under §1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Date: ____________ ______________________________
Liang-Jie Zhang

Date: ____________ ______________________________
John Youssef Sayah

Date: ____________ ______________________________
Tian-Jy Chao

Date: ____________ ______________________________
Ying Nan Zuo

Date: April 1, 2009 ______________________________
Shun Xiang Yang

Date: April 1, 2009 ______________________________
Jing Min Xu

Date: ____________ ______________________________
Haiyan Wang, signing on behalf of
deceased inventor Yiming Ye

EXHIBIT 1

METHOD AND APPARATUS OF INFORMATION HYPERCHAIN MANAGEMENT FOR ON-DEMAND BUSINESS COLLABORATION

DESCRIPTION

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

The present invention generally relates to systems for information hyperchain management for on-demand business collaboration. Nowadays, a business entity is not standalone anymore and does not need to produce everything it needs. It can outsource portion of its business to its partners. It is the outsourcing model that enables disaggregated businesses to form a value chain for creating more innovative and higher quality products or services than would have otherwise accomplished by itself. In the engineering design collaboration scenario, many manual or semi-automated operations exist in the design process. The real challenges when enterprises are collaborating together are ability to retrieve information in an On-Demand fashion and to monitor and manage the status of projects, tasks, and exchanged documents in real-time. This invention solves the above mentioned business problems and enables disaggregated business entities to reduce cost and respond with speed to market changes and business needs.

BACKGROUND DESCRIPTION

Nowadays, enterprises are not standalone anymore. Each enterprise needs to work with its logistic service providers and customers to keep the enterprise running well. An enterprise does not need to be a kingdom which produces everything needed for its "island". It leverages some well-proven services and products to be part of its daily operation process or a new product design process. Let's take a new product design as an example, the enterprise that is designing products have to work with the components suppliers, electronic manufacturing services (EMS), or contract manufactures to design a product collaboratively. Some component design work may be outsourced to design partners who are specializing at a special component such as ASIC chip, Battery, or Motherboard. Actually, this kind of outsourcing model even becomes more and more popular in the IT industry and business transformation industry. It is the outsourcing model that enables disaggregated businesses to form a value chain for creating more innovative and higher quality products or services than that they would have accomplished by their owns.

In a typical value chain, the trading partners or design partners could be dynamically added or removed in the lifecycle of an e-business solution. All the resources including business entities, services provided by business entities, documents and messages should be kept in an on-demand model. That is, when you need it, you can use it directly or download a piece of code to access the available resources remotely. This on-demand fashion provides a way for disaggregated business entities to perform business interactions in an efficient and cost-effective way.

The real technical obstacles are not the business process representation and data transformation. The real problem is the interactions between two or more loosely coupled business processes, which could be private business processes in their own enterprises or public processes across multiple enterprises. For example, in an engineering design collaboration scenario, there are still manual or semi-automated operations in the product design process. They use phone, fax, or email to exchange design specification documents and design files such as electronic CAD file, mechanical CAD file as well as Bill of Materials (BOM) files. The resulting problem is that it is very hard to get the real-time information from the design network such as monitoring the status of the on-going projects, tasks, and exchanged documents. Additionally, there are different design systems and product development management (PDM) systems with different formats for design documents and design specifications. Moreover, the traditional workflow only documents the detailed steps of a known process. But for a product innovation, nobody really knows all the details of the design for each individual component. Therefore, it is a non-deterministic workflow that involves multiple collaborators who are specializing in their domain components. Especially, lots of business exceptions need to be addressed during the full life cycle of a product design process. Currently, there is no standard to support this kind of engineering design collaboration, which is listed in the white space of RosettaNet standard specification. As a result, there is a pain point is the low efficiency for information exchange due to the manual or semi-automated operations.

In the current business to business application scenarios, tons of documents have to be transmitted to the receivers in different enterprises. Theoretically, the documents should be transmitted in an efficient and controlled way. That means, the documents should be delivered to the right people with appropriate responses. Meanwhile, the status of the transmitted documents should be easily monitored in a distributed environment. In fact, for each document, it can be delivered to any receivers. After one of the receivers gets this document, he or she may modify this document and send back to the sender or distribute to other partners. From the original sender's point of view, he or she may not know who else will be the receivers. So the document delivery channel across multiple enterprises or even within one organization actually forms a nonstructural and non-deterministic information exchange flow. Along with the control information which could be created by the sender and re-created by the receivers for further distribution, the nonstructural and non-deterministic information exchange flow results in hierarchical linkages.

Therefore, with the rapid growth of the requirements of business process integration, how to efficiently and effectively manage the nonstructural and non-deterministic information exchange in a uniform way is becoming a challenging issue. More specifically, we are facing the following problems today:

1). Lack of a uniform annotation mechanism that represents the control information separated from the real documents to be exchanged;
2). Lack of a comprehensive collaboration pattern that covers all aspects of a nonstructural and non-deterministic information exchange flow;
3). Lack of an efficient on-demand information mechanism that dynamically locates resources from the nonstructural relationship graph as well as intelligently aggregates data from multiple sources;
4). Lack of an trackable information embedding mechanism for visibility control that addresses key issues associated with data exchange in collaborative environment;

To provide an on-demand information exchange model, tailored to an individual recipient as well as support the capability of monitoring and tracking of the information delivered and exchanged, this disclosure proposes a HyperChain annotation technology referred as HyperChain based On-Demand Information Exchange Mechanism (ODIEM). The HyperChain Annotation technology extends the concept of Hyperlink in the individual objects in HTML files to business processes collaboration area, supporting documents exchange for heterogeneous data streams in outsourcing, process tracking, visibility control, and the like. HyperChain annotation data can be used by the users by following the link and downloading the documents on need basis. It's basically a hierarchical annotation linkage. Every document in the business collaboration environment is annotated by certain properties. HyperChain annotations are exchanged before any documents exchanges. The receiver of HyperChain annotations can take actions accordingly so that document exchanges are on demand based, which undoubtedly can reduce unnecessary communication traffic among the business collaboration community.

The HyperChain based On-Demand Information Exchange Mechanism (ODIEM) aims at enabling the e-business solution to automate the dynamic configuration and instantiation of business processes using predefined collaboration primitives and business constructs. In addition, it provides a foundation to support flexible, distributed, secure peer-to-peer architecture of business collaboration (e.g. B2B) without the need of a central hub.

We use an engineering design collaboration scenario as an example to describe the HyperChain based information management for generic business collaboration. As we said earlier, with the rapid growth of the requirements of business process integration, efficiently and effectively manage the nonstructural information (e.g. un-predefined structure of information representation) exchange is an important issue.

In this disclosure, we propose a new type of annotation representation using Resource Description Framework (RDF) [1] referred as "HyperChain RDF" for annotating a chain of design data, or design chain, the associated design documents, and the associated actions. RDF is a modeling system with facilities for addressing metadata exchange and it is near scheme-less. Therefore, benefit of using HyperChain RDF is that there is no need to predefine schema for each application because it can accommodate data of various format without having a fixed, predefined schema. Therefore, it is flexible and suitable for annotating different data types needed for business collaboration, such as design data, specifications, requirements, Bill Of Material (BOM), as well as partner profiles, business processes such as reviewing, designing and project evaluation; access controls; collaboration patterns,; as well as actions for status tracking, file exchange, and so on.

SUMMARY OF THE INVENTION

This invention includes a method and apparatus, comprising the individual and aggregate use of the following techniques:

1. HyperChain Annotation - Enable interaction between loosely-coupled business processes or interacting partners and business entities, Similar to hyperlinks of HTML, the annotated information components and process components for collaboration are expressed as links
2. On-demand Message Exchange – data are not required to be aggregated before send. Instead, data can be later retrieved on need basis, i.e. send fragmented data
    a. Send schema-less Hyperchain annotation data first
    b. Fetch the detailed information like opportunity data, project management notifications, design files, design specifications, BOM files based on the roles of the receiver (on-demand data access/transfer)
3. Status/state information embedded in messages transmitted - Enable status tracking both for collaborative processes, i.e. project/product management, and for documents, i.e. design files, BOM files, etc.

4. Flexible Collaborative business message exchange patterns comprising Collaborative business constructs, which consists of multiple collaborative business primitives and business constructs:
    a. Payload information can be opaque and need not be pre-defined types
    b. Enable standardized way to contain non-standard data, i.e. allowing multiple bi-lateral collaborations without having to go through standard bodies
5. Collaborative Directory – ontology persistent storage for partner, project and service profiling,
    a. Access control requirement may utilize identity management directory integration functions

Web Services enabled utilities in Collaborative Directory

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages will be better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, in which:

Figure 1 is a diagram showing the architecture of a distributed peer to peer collaboration.
Figure 2 is a diagram showing the diagram of HyperChain based On-Demand Information Exchange Model.
Figure 3 is a diagram showing an example Design HyperChain.
Figure 4 is a diagram showing hyperlinking annotation.
Figure 5 is a diagram showing the HyperChain Annotation Data Graph for Request for Design (RFD).
Figure 6 is a diagram showing an example of collaboration primitive and business construct.
Figure 7 is a diagram showing outsourcing scenario composed by business constructs.
Figure 8 is a diagram showing HyperChain Manager.
Figure 9 is a screen showing annotation creation for a resource.
Figure 10 is a screen showing creating an outsourcing task.
Figure 9 is a screen showing annotation creation for design requirements.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT OF THE INVENTION

This disclosure is organized as follows: First, there is described the main components of the systems, followed by key mechanisms used to enable the extended business collaboration; next, there is a detailed structure of the ontology and the on-demand information delivery model for extended business collaboration. A embodiment is described as well as Patent enforcement/detectable features. Last, a case study is presented with the framework embodiment of a prototype.

Figure 1 shows an example distributed solution architecture, leveraging Web Services technologies, gateway software for business process integration across multiple enterprises.

Initially, the documents to be exchanged are annotated with hyperlinks that point to location where the real data can be downloaded or exchanged, and the hyperlinks are forming a logical chain of actions that the recipients can take to further delve into the details of the information and download them on need basis. That is, instead of bombarding the recipients will levels of documents to be exchanged all at once, the sender will first send the annotations only. Once the recipient receives it, he or she can follow the hyper chain and take action according to their roles and authorizations in the enterprises, such as Project Manager, Design Engineer or Purchasing Manager, to download only those data that are needed.

In this disclosure, we design our on-demand information exchange model to meet the following design goals:

A. Provide a flexible and uniform annotation representation for information exchange of various non-structured data without requiring pre-defined schemas.
B. Automate the annotation data generation process.
C. Capture and automate business collaboration interaction patterns for information exchange based on the annotation data.

Taking the above goals into consideration, we present a HyperChain based On-Demand Information Exchange Mechanism (ODIEM) that paves the way for supporting flexible, distributed, secured and controlled peer-to-peer architecture of business collaboration (e.g. B2B) without the need of a central hub. The solution architecture is shown in Figure 2.

The major components of this solution architecture are as follows:
(1) Comprehensive Collaboration Pattern - iterative actions taken place amongst the enterprises that collaborate on certain business processes
(2) Flexible Schema-less Annotation Structure - data format of the annotations
(3) Annotation Storage - stores annotation data for visibility control, information delivery and management
(4) GUI based Annotation Creation - mechanisms to capture, create and automate the annotations
(5) Effective HyperChain Manager - Key component that creates and manages annotations

In addition, it is the delivery policy that describes how the on-demand contents are to be delivered. There are at least three types of delivery mechanism:

(1) Scheduled content delivery - On a predetermined, periodical schedule, the information content can be delivered to the recipients

(2) On-demand content delivery - Ad-hoc, based on user's request, i.e. clicking on the hyperchain links that provided with the annotation data and download the data as well as trigger a file transfer service to move file(s) from one place to another place.

(3) Access control-based content delivery - delivering contents depending upon the role and authorization of a recipient.

The basic idea of the On-demand Message Exchange Model is to send schema-less HyperChain annotation data first, then for the recipients to follow the hyper chains to fetch the detailed information like design files, design specifications, BOM files based on the roles of the recipient (on-demand file transfer) through self-retrieving or agent-based file transfer service . In the mean time, it supports tractable information associated with design files, design process and BOM.

The sections that follow describe in details each of the four major components in this proposed solution architecture.

2.1 On-Demand Business Collaboration Ontology

An ontology or commonly shared knowledge defines the business semantics to annotate information to be exchanged, and it provides the collaboration foundation. Without such shared common knowledge, participants will not be able to decipher the exchanged information. Current collaboration solutions are usually based on fixed knowledge pre-configured at each collaborator side, which reduces the collaboration flexibility and extensibility.

The business collaboration ontology is based on a basic ontology across multiple industries. In this disclosure, all the collaborators use the basic ontology to exchange business information. The business collaboration ontology uses RDF model to specify, and annotation is one part of the ontology. For example, we defined RDF resources such as “Site”, “Organization”, “Project”, “Task”, “Requirement”, “Transaction”, “Documents”, “Annotation”, etc. in a RDF schema. The RDF schema serves as the basic ontology definition that all collaborators need to understand.

We propose the basic resources for generic business collaboration and some extensions for engineering design collaboration. The Business Collaboration Ontology is extensible for additional annotations. Collaborators can define their own ontology and add additional annotations into the basic ontology. Collectively, the basic and extended ontologies become one entire ontology that we uses to create resources and model. Collaboration ontology provide a flexible and uniform annotation representation for information exchange of various non-structured data without requiring pre-defined schemas In addition, with annotated status and access control as part of the collaboration primitive, processes as well as documents exchanged can be tracked, monitored and controlled.

Since RDF-based business collaboration ontology treats all the new added entities as resources, we will use the same mechanism to handle those new added resources as the way to handle the existing ones. The extensive and flexible features of the business collaboration ontology can allow you define any annotations without worrying about the schema of the annotation data. That’s why we say the HyperChain annotation data that conforms to business collaboration ontology is schema-less.

2.2 Annotation Structure

2.2.1 Basic Annotation for Business Collaboration Chain

The annotation data is used to annotate the entire or partial business collaboration across multiple organizations. They provide links to distributed information and address key issues associated with data exchange in collaborative environment by specifying Content, Structural, Routing, Location, Life Time/Persistence as well as Availability and Viability with respect to collaborative processes. Annotation data of various resources for business collaboration need to be created such as via graphic user interface or an application and stored in persistence storage before they can be retrieved to annotate exchanged information at runtime. As shown in Figure 3, the design hyperchain diagram illustrates the annotation data which are related to and parallel with the upstream and downstream data.

To effectively annotate the data for extended business collaboration, Collaborative eXchange Protocols (CxP) use a hierarchical, top-down approach for the annotation data. For example, in the engineering design collaboration scenario, based on the design process model, annotation data are divided into hierarchical representations, i.e. starting with annotations for design collaboration processes, followed by annotations for design activities specific to CxP, and, then, annotations for design tools used by the business entities involved in the engineering design collaboration scenario.

In the design collaboration scenario, the hierarchical annotations are employs as follows. Business process level annotation serves as an overall message for lower level Design Collaboration Primitive. It instructs the recipient to take some actions in response to the annotation data. For example, when design partner A wants to send a Request for design (RFD) to design partner B, he or she sends out a business process annotation message with a tag indicating it is for an Request for design. Upon receiving it, partner B takes corresponding actions based on the information received from the business process annotation; the actions may be to use the design file's design activity annotation link to retrieve more info about design requirement, or use the design process's design activity annotation link to get more info about the design process, etc. The 'process' to handle the annotation data varies, probably based on the key information, e.g. OP , in annotation. Therefore, for each primitive, there are one or more interactions between the collaborators, i.e. one business process annotation message sent first, followed by different-level information requests and/or responses, organized in a sequence.
After receiving the business process annotation, partners can view the annotation and merge it with their own annotation storage. If a partner wants to know more about one of the annotated resources, he can get the annotation links, e.g. design file annotation link, and request for more information. The sender will generate annotation for the design file and send it back to the partner. The annotation for design files and tool annotations serve as the Data Holder for design collaboration. They instruct the recipient to take actions using some "data" in a certain way. The partner can determine whether or not to retrieve the actual design file based on the annotation.

The difference between design activity annotation and tool annotation is that: design activity annotation data are specific to a extended business collaboration infrastructure and common for all design activities; tool annotation data are specific to various design tools, which is transparent to the extended business collaboration infrastructure.

2.1.2. *Annotation for information exchange flow*

As shown in Figure 5, the design collaboration diagram using RDF model (or RDF graph) illustrates the annotation data model, for Request for Design message (RFD) interchange protocol and defines several properties for information exchange.

Based on the above annotation graph, an XML file or data stream defines an RDF instance to capture the annotation information associated with the Request for Design (RFD) protocol, such as design files, design process as well as access control information. Note that the XML data stream can be stored in relational database.

The following example shows a sample business process annotation data instance for RFD message of the entire design chain. The "transaction" resource defines the exchange context of the messages, RFDTransaction, the requester, YDT, and the responder, MyComputerCorp. The. "Task", T61MotherBoardDesign, and "Project", T61BoardDesign, resources define which project and task the messages are bound to. Several containers are defined to group the following annotations or metadata: Design specification annotation, T61Specification.pdf, Design file annotation, T21MotherBoardDesignFile.cat, BOM file annotation, T21BOMFile.bom, Design process annotation, outsourcingConstraint, and other related annotations, such as access control.

```
<rdf:RDF
<cbpm:RFD rdf:ID="007">
      <cbpm:transaction>
      <cbpm:Transaction
rdf:about= »http://www.ydt.com/dc/directory/transaction#007001 »>
      <rdf:type>&cbpm;RFDTransaction</rdf:type>
      <cbpm:requester
rdf:resource="http://www.ydt.com/dc/directory/Organization#YDT"/>
      <cbpm:responder
rdf:resource="http://www.ydt.com/dc/directory/Organization#MyComputerCorp"/>
      <!--anything about the Transaction-->
      </cbpm:Transaction>
      </cbpm:transaction>
      <cbpm:task>s
      <cbpm:Task
rdf:about=“http://www.ydt.com/dc/directory/task#T61MotherBoardDesign”>
      <!--anything about the task-->
      <cbpm:forProject>
      <cbpm:Project
rdf:about=“http://www.ydt.com/dc/directory/project#T61BoardDesign”>
      <!--anything about the project-->
      </cbpm:Project>
      </cbpm:forProject>
      </cbpm:Task>
      </cbpm:task>
      <cbpm:requirement>
      <cbpm:Requirement
rdf:about=“http://www.ydt.com/dc/directory/requirement#00d034334”>
      <!--anything about the requirement-->
      <cbpm:specification>
      <rdf:Bag>
      <rdf:li>
      <cbpm:Specification
rdf:about=“http://www.ydt.com/pdf/T61Specification.pdf”/>
      </rdf:li>
      </rdf:Bag>
      </cbpm:specification> <cbpm:reference>
      <rdf:Bag>
      <rdf:li>
      <cbpm:DesignFile
rdf:about=“http://www.ydt.com/pdf/T21MotherBoardDesignFile.cat”/>
      </rdf:li>
      <rdf:li>
      <cbpm:BOMFile rdf:about=“http://www.ydt.com/pdf/T21BOMFile.bom”/>
      </rdf:li>
      </rdf:Bag>
      </cbpm:reference>
      <cbpm:designProcess>
      <rdf:Bag>
      <rdf:li>
      <cbpm:OutsourcingConstaints
rdf:about=“http://www.ydt.com/outsourcingConstraint”/>
```

```
    </rdf:li>
    <rdf:li>
    <cbpm:AccessControl rdf:about="http://www.ydt.com/accessControl"/>
    </rdf:li>
    </rdf:Bag>
    </cbpm:designProcess>
    </cbpm:Requirement>
    </cbpm:requirement>
    </cbpm:RFD>
</rdf:RDF>
```

2. 1.3. *Design Activity Annotation*

The design activity annotations mainly focus on constraints in the design collaboration process. There are several types of activity annotations, i.e. design requirements, design configurations/specifications, the design files, BOMs, design processes, etc., and each one is for a different purpose. Thus, each contains different annotation data with a different format. However, they should all follow the same design rule.

In Example below, a sample activity design file is shown. The information might include the following: design requirements, design configuration/specifications, the designed files, and tools, etc.

```
<cbpm:DesignFile
rdf:about="http://www.nec.com/AmpSubMinDconn9PosRearPanelMtg.CATPart">
<cbpm:fileName> Amp Sub Min D conn 9 Pos Front Panel Mtg.CATPart
</cbpm:fileName>
<cbpm:fileSize> 239K bytes </cbpm:fileSize>
<cbpm:lastModifiedTime>9/25/2002, 6:00PM </cbpm:lastModifiedTime>
<cbpm:lastVersionNumner> V5.1 </cbpm:lastVersionNumner>
<cbpm:location>
ftp://ftp.nec.com/dc/designfile/AmpSubMinDconn9PosRearPanelMtg.CATPart
</cbpm:location>
<cbpm:designTool> CATIA V5 </cbpm:designTool>
<cbpm:format> CATPart </cbpm:format>
<cbpm:designPartnerID> NEC </cbpm:designPartnerID>
```

```
<cbpm:designProjectID> ThinkPadT61 </cbpm:designProjectID>
<cbpm:accessControl>
    ... ...
</cbpm:accessControl>
</cbpm:Specification>
```

The following example shows a design activity annotation where several constraints are specified as well as the access control using OASIS eXtensible Access Control Markup Language (XACML) [2] to express the constraints.

```
<cbpm:design-activity-annotation>
<cbpm:desc>
<cbpm:checkpointConstraints>. . . </cbpm:checkpoint constraints>
<cbpm:outsourcingConstraints>... (XACML) </cbpm:outsourcingConstraints>
<cbpm:acl>  ... (XACML) </cbpm:acl>
<cbpm:documentFormatConstraints>http://temporg.com/tempuri/documentFormatCont
raints.htm</cbpm:documentFormatConstraints>
      </cbpm:desc>
</cbpm: design-activity-annotation>
```

2. 1.4. Tool Annotation

Tool annotations are for design tools, and they are platform-dependent. Some examples of such tool annotations about a BOM file are Processing Platform, designStatus (on-going or redesign), review status, modification history, password to download, etc.

```
<cbpm:design-process-low-level-annotation>
<cbpm:bomDesc>
<cbpm:fileProperty>
            <cbpm:filename>CDRom</cbpm:filename>
            <cbpm:fileSize>1020K</cbpm:fileSize>
            <cbpm:creationTool>MSEXCEL</cbpm:creationTool>
            <cbpm:documentId>ThinkPad-Bom-010</cbpm:documentId>
      </cbpm:fileProperty>
</cbpm:processingPlatform>
<cbpm:DevelopmentPlatformID>UNIX</cbpm:DevelopmentPlatformID/>
<cbpm:ParentDesignToolID>CATIAPP</cbpm:ParentDesignToolID/>
<cbpm:AccessibleParty>http://temporg.com/tempuri/acl/ ThinkPad-Bom-010AccessList</cbpm:AccessibleParty>
      </cbpm:processingPlatform>
```

```
</cbpm:bomDesc>
  </cbpm:tool-annotation>
```

2.2. Annotation Storage

The design collaboration ontology is defined in RDF schema format, and is stored in RDF format. Annotation is one part of the ontology.

There're diverse requirements for annotation in design collaboration, and new requirements emerge in endlessly. In addition to the pre-defined annotations for eBC, users can define custom annotations. Following is a sample of the storage of the annotation definition:

```
<rdf:RDF
  xmlns:daml='http://www.daml.org/2001/03/daml+oil#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'>
  <daml:DatatypeProperty rdf:about='http://www.ibm.com/ibm/ydt#fileName'
     rdfs:label='fileName'>
    <rdfs:comment>The name of a file, huh?</rdfs:comment>
    <rdfs:domain rdf:resource='http://www.w3.org/2000/01/rdf-schema#Resource'/>
    <rdfs:range rdf:resource='http://www.w3.org/2000/10/XMLSchema#string'/>
    <rdfs:isDefinedBy rdf:resource='urn:Organization:YDT@YDT'/>
  </daml:DatatypeProperty>
  <daml:DatatypeProperty rdf:about='http://www.ibm.com/ibm/ydt#fileSize'
     rdfs:label='fileSize'>
    <rdfs:comment>The size of a file</rdfs:comment>
    <rdfs:domain rdf:resource='http://www.ibm.com/cbpm#Document'/>
    <rdfs:range rdf:resource='http://www.w3.org/2000/10/XMLSchema#string'/>
    <rdfs:isDefinedBy rdf:resource='urn:Organization:YDT@YDT'/>
  </daml:DatatypeProperty>
  <daml:DatatypeProperty rdf:about='http://www.ibm.com/ibm/ydt#format'
     rdfs:label='format'>
    <rdfs:comment>The format of a file, huh?</rdfs:comment>
    <rdfs:domain rdf:resource='http://www.ibm.com/cbpm#Document'/>
    <rdfs:range rdf:resource='http://www.w3.org/2000/10/XMLSchema#string'/>
    <rdfs:isDefinedBy rdf:resource='urn:Organization:YDT@YDT'/>
  </daml:DatatypeProperty>
</rdf:RDF>
```

These annotation definitions are applied to various elements during the information exchanges in design collaboration processes. Following is a storage (in RDF representation) of the annotated information during the RFD primitive for a design project.

```
<rdf:RDF
```

```
  xmlns:RDFNsId1='http://www.ibm.com/ibm/ydt#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:RDFNsId2='http://www.ibm.com/cbpm#'>
  <RDFNsId2:RFD
rdf:about='urn:RFD:task1_RFD1@task1_RFD1@CPUDesign@SuperComputer@YDT@Y
DT'
      RDFNsId2:identifier='task1_RFD1'
      RDFNsId2:description=''
      RDFNsId2:status='Received'>
    <RDFNsId2:creationTime>Dec 10, 2002 8:12:21 PM</RDFNsId2:creationTime>
    <RDFNsId2:transaction>
      <RDFNsId2:RFDTransaction
rdf:about='urn:Transaction:task1_RFD1@CPUDesign@SuperComputer@YDT@YDT'
          RDFNsId2:status='Created'
          RDFNsId2:identifier='task1_RFD1'>
        <RDFNsId2:requester>
          <RDFNsId2:Organization rdf:about='urn:Organization:YDT@YDT'
              RDFNsId2:identifier='YDT'
              RDFNsId2:schemaExtensionURL='http://www.ibm.com/ibm/ydt'>
          </RDFNsId2:Organization>
        </RDFNsId2:requester>
        <RDFNsId2:creationTime>Dec 10, 2002 8:12:21 PM</RDFNsId2:creationTime>
        <RDFNsId2:forTask>
          <RDFNsId2:OursourcingTask
rdf:about='urn:Task:CPUDesign@SuperComputer@YDT@YDT'
              RDFNsId2:identifier='CPUDesign'
              RDFNsId2:status='RFDReceived'>
            <RDFNsId2:performer>
              <RDFNsId2:Organization rdf:about='urn:Organization:NEC@NEC'
                  RDFNsId2:schemaExtensionURL='http://www.ibm.com/ibm/nec'
                  RDFNsId2:identifier='MYASIC'>
              </RDFNsId2:Organization>
            </RDFNsId2:performer>
            <RDFNsId2:description>I am designing the fastest CPU in the
world</RDFNsId2:description>
            <RDFNsId2:name>Designing a CPU</RDFNsId2:name>
            <RDFNsId2:forProject>
              <RDFNsId2:Project rdf:about='urn:Project:SuperComputer@YDT@YDT'
                  RDFNsId2:status='Created'
                  RDFNsId2:identifier='SuperComputer'>
              </RDFNsId2:Project>
            </RDFNsId2:forProject>
            <RDFNsId2:creationTime>Dec 10, 2002 8:12:21
PM</RDFNsId2:creationTime>
          </RDFNsId2:OursourcingTask>
        </RDFNsId2:forTask>
        <RDFNsId2:responder rdf:resource='urn:Organization: MYASIC @ MYASIC />
      </RDFNsId2:RFDTransaction>
    </RDFNsId2:transaction>
    <RDFNsId2:requirement>
      <RDFNsId2:DesignRequirement
rdf:about='urn:DesignRequirement:req1@CPUDesign@SuperComputer@YDT@YDT'
         RDFNsId2:description='desc'
         RDFNsId1:TFTSize='14.1'
         RDFNsId2:identifier='req1'
         RDFNsId1:cpuFrequency='2.5G'>
        <RDFNsId2:name>requirement 1</RDFNsId2:name>
        <RDFNsId2:reference>
          <rdf:Bag>
            <rdf:li>
              <RDFNsId2:DesignFile
rdf:about='urn:Document:Ref1@req1@CPUDesign@SuperComputer@YDT@YDT'
                 RDFNsId1:fileAuthor='robert'
                 RDFNsId1:fileName='cpurequirementRef1.doc'
                 RDFNsId2:identifier='Ref1'>
                <RDFNsId2:name>ref 1</RDFNsId2:name>
                <RDFNsId2:description>This is ref 1</RDFNsId2:description>
              </RDFNsId2:DesignFile>
            </rdf:li>
          </rdf:Bag>
        </RDFNsId2:reference>
        <RDFNsId2:specification>
          <rdf:Bag>
            <rdf:li>
              <RDFNsId2:Specification
rdf:about='urn:Document:Spec1@req1@CPUDesign@SuperComputer@YDT@YDT'
                 RDFNsId1:fileAuthor=‘robert’
                 RDFNsId2:identifier='Spec1'
                 RDFNsId1:fileName='cpurequirement.doc'>
                <RDFNsId2:description>This is doc 1</RDFNsId2:description>
                <RDFNsId2:name>doc 1</RDFNsId2:name>
              </RDFNsId2:Specification>
            </rdf:li>
          </rdf:Bag>
        </RDFNsId2:specification>
        <RDFNsId2:forTask
rdf:resource='urn:Task:CPUDesign@SuperComputer@YDT@YDT'/>
      </RDFNsId2:DesignRequirement>
    </RDFNsId2:requirement>
  </RDFNsId2:RFD>
  <RDFNsId2:ReceiptAck
rdf:about='urn:ReceiptAck:task1_RFD1ReceiptAck@task1_RFD1@CPUDesign@SuperCo
mputer@YDT@YDT'
      RDFNsId2:identifier='task1_RFD1ReceiptAck'
      RDFNsId2:status='Created'>
     <RDFNsId2:transaction
rdf:resource='urn:Transaction:task1_RFD1@CPUDesign@SuperComputer@YDT@YDT'/>
     <RDFNsId2:creationTime>Dec 10, 2002 8:04:02 PM</RDFNsId2:creationTime>
  </RDFNsId2:ReceiptAck>
  <RDFNsId2:Individual rdf:about='urn:Individual:Mike@YDT@YDT'
      RDFNsId2:identifier= Mike >
     <RDFNsId2:memberOf rdf:resource='urn:Organization:YDT@YDT'/>
     <RDFNsId2:description>This is an user.</RDFNsId2:description>
     <RDFNsId2:name> Mike Josh</RDFNsId2:name>
  </RDFNsId2:Individual>
</rdf:RDF>
```

The annotation can be stored in a Collaborative Directory, which can be deployed on each enterprise site. Since the data with status information about the project, task, exchanged documents, and etc. are stored in multiple collaborative directories, we can aggregate the use information from these distributed collaborative directories based on access control policy carried in the annotation data.

2.3 Annotation Creation

Annotation creation is a major function of the On-demand Information Exchange Mechanism. As mentioned above, all the annotation data of various resources used in business collaboration are stored in annotation storage such as plain text file or relational database. The annotation creation process should operate on the storage to create annotations. For one embodiment, the creation of annotation includes the following two steps:

(1) Collect information by use of extended business collaboration portal. The information includes the description of various resources such as partners, projects, tasks, specification annotations, reference design file annotations, other related annotations and so on.
(2) Store all the collected information into the annotation storage.
(3) Extract required data from the storage to organize the annotation message to be exchanged

Let us take the RFD message creation as an example to illustrate the process.

First, user should create a new task as an outsourcing task or internal task shown in Figure 10.

Then, he should follow the following steps to specify design requirements for the design task. The requirements include specifications, reference design files, design process constraints and so on, which are illustrated in Figure 11.

After the information is collected and stored in the annotation storage, annotation creation process starts. The annotation creation module extracts required data from the storage and forms the RFD annotation message based on RFD message data model. The generated RFD message will be sent to design partners.

After receiving the RFD Message, partners can view the annotation and merge it with their own annotation storage. If a partner wants to know more about one of the annotated resources, he can get the annotation link (such as design file annotation link) and request for more information. The sender will generate an annotation for the design file and send it back to the partner. The partner can determine whether or not to retrieve the actual design file based on the annotation. Thus, on-demand information exchange is performed.

2.4 Collaboration Pattern

CxP uses Resource Definition Framework (RDF) to annotate design collaboration processes by defining industry specific ontology, allowing peer-to-peer interaction between collaborative processes. RDF is a general-purpose language and common framework for representing metadata (or properties) of Web resources and it is defined in XML syntax. CxP builds on top of a set of standard protocols and adds the features needed for extended business collaboration processes. The overall protocol architecture can be described in Figure 6.

In Messaging Layer RDF is used to represent business collaboration annotation which is called HyperChain annotation. On top of that, we define a set of primitives as Collaboration primitives to help the communication and collaboration between the parties.

A Business Construct is a basic unit of message exchange sequences which serve a single business goal. For example, RFD business construct is used when a request for design is initialized, i.e., A design center. Yamato Design Team shown in Figure 8, can send RFDs to its design partners to do motherboard design or to do mechanical and electrical design. Following that Accept or Reject primitive may be received from the design partners. Later a design partner may submit its design results by DS primitive.

A Business Scenario is serving a more complex business goal like Early Design-In scenario. Each Business Scenario may consist of several business constructs depending on the corresponding business context.

The details of the collaboration primitive, business construct, and business scenario are described below.

Based on these protocols, collaborators can define the DC processes as they want. Following is a ThinkPad design work process, which involves multiple collaborators and is composed of multiple protocols. The figure shows only the fragment, which is between YDT and NEC, of the whole process.

In CxP, an atomic message is defined as an rudimentary exchanges of information between trading partners, e.g. an RFD message. A set of choreographed messages forming a primitive, e.g. RFD primitive, comprising two messages, i.e. RFDMessage and AckMessage. Furthermore, one or more primitives form a Business Construct, e.g. RFD Business Construct, comprising two primitives, i.e. RFD primitive and Acceptance/Rejection primitive. Scenarios are sequences of business constructs that represent a complex interaction among business partners, such as Design initialization, Engineering Change Management, and Opportunity Launch. In addition CxP primitives and business constructs are targeted for specific collaboration goals, even though configurable, are relatively fixed, while business scenarios can be composed in many ways and thus quite flexible.

A design collaboration primitive is a group of message exchange for a specific and micro design collaboration goal. Several core design collaboration primitives are defined for CxP:

- RFD (Request For Design)
- Accept/Reject (Accept or Reject a request)
- DS (Design Submission)
- RFI (Request For Information)
- IS (Information Submission)
- RFU (Request For Update)
- US (Update Submission)

Let’s take RFD as an example; each collaborator uses the RFD Primitive to request a partner to perform a design task. An RFD primitive comprises three messages: RFD, RFD_Receipt_Ack, and RFD_Acceptance_Ack messages.

- RFD Message: sent by the originator, e.g. a design center, to a recipient, e.g. design partner. Contains a requirement comprising specifications, references, and annotations.
- RFD_Receipt_Ack Message: sent by the recipient; a response to RFD message, indicating the RFD message has been received by the recipient.

```
<RDFNsId2:RFD
rdf:about=’urn:RFD:SoundCard_test1@SoundCard_test1@SoundCard@Workstation@YDT@
YDT’
RDFNsId2:description=”
RDFNsId2:identifier=’SoundCard_test1’
RDFNsId2:status=’Accepted’>
<RDFNsId2:requirement>
```

```
<RDFNsId2:DesignRequirement
rdf:about=‘urn:DesignRequirement:test1@SoundCard@Workstation@YDT@YDT’
RDFNsId1:cpuFrequency=‘500’
RDFNsId2:identifier=‘test1’
RDFNsId2:name=‘test1’
RDFNsId2:description=‘test1’>
<RDFNsId2:specification>
<rdf:Bag>
<rdf:li>
<RDFNsId2:Specification
rdf:about=‘urn:Document:test@test1@SoundCard@Workstation@YDT@YDT’
RDFNsId2:identifier=‘test’
RDFNsId2:name=‘test’
RDFNsId2:description=‘d:\there’/>
</rdf:li>
<rdf:li>
<RDFNsId2:Specification
rdf:about=‘urn:Document:@test1@SoundCard@Workstation@YDT@YDT’
RDFNsId2:description=‘’
RDFNsId2:name=‘’
RDFNsId2:identifier=‘’/>
</rdf:li>
</rdf:Bag>
</RDFNsId2:specification>
<RDFNsId2:reference
rdf:type=‘http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag’/>
<RDFNsId2:forTask rdf:resource=‘urn:Task:SoundCard@Workstation@YDT@YDT’/>
</RDFNsId2:DesignRequirement>
</RDFNsId2:requirement>
<RDFNsId2:transaction
rdf:resource=‘urn:Transaction:SoundCard_test1@SoundCard@Workstation@YDT@YDT’/
>
<RDFNsId2:creationTime>Jan 16, 2003 5:10:17 PM</RDFNsId2:creationTime>
</RDFNsId2:RFD>
```

Each design partner may accept or reject the request after received either a RFD or RFU. One example of Accept primitive to a RFD is as the following:

```
<RDFNsId2:AcceptanceAck
rdf:about='urn:AcceptanceAck:SoundCard_test1AcceptanceAck@SoundCard_test1@Sou
ndCard@Workstation@YDT@YDT'
RDFNsId2:identifier='SoundCard_test1AcceptanceAck'
RDFNsId2:ack='Accept'
RDFNsId2:status='Sent'>
<RDFNsId2:transaction
rdf:resource='urn:Transaction:SoundCard_test1@SoundCard@Workstation@YDT@YDT'/
>
<RDFNsId2:creationTime>Jan 16, 2003 5:38:38 PM</RDFNsId2:creationTime>
</RDFNsId2:AcceptanceAck>
```

Business Construct

A business construct comprises a group of collaboration primitives, which can be selectively configured for a business construct. Once configured, a business construct is organized in a relatively fixed fashion to achieve a single design collaboration goal. The following business constructs are based on the primitives discussed before:

- RFD business construct (RFD primitive + Accpet/Reject primitive+ DS primitive)
- RFU business construct (RFU primitive + US primitive)
- RFI business construct (RFI primitive + IS primitive)
- US business construct (US primitive)
- IS business construct (IS primitive)

Based on these business constructs, collaborators can define any complex business scenario as they want.

The business constructs are often very complicated because they often involve multiple interactive messages based on the collaboration primitives.

A Standard protocol for business process modeling language, such as BPEL4WS [3], can be used to represent CxP business constructs and business scenario. Once represented by BPEL4WS, multiple business constructs can form a business scenario, which can be dynamically composed.

Example: RFD business construct

A RFD business construct may contain one RFD primitive, one Accpet/Reject primitive, and one DS primitive. The RFD micro flow can be represented using BPEL4WS in the following:

```
<process name="RFDmicroflow"
        targetNamespace="urn:samples:BusinessConstructs"
        xmlns:tns="urn:samples:BusinessConstructs"
        xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/">
 <partners>
        <partner name="RFDoriginator"
                serviceLinkType="tns:RFDoriginatingSLT"
                myRole="RFDoriginating"/>
        <partner name="RFDreceiver"
                serviceLinkType="tns:RFDreceivingSLT"
                myRole="RFDreceiving"/>
        <partner name="buyer"
      serviceLinkType="tns:buyingSLT"
      myRole="buying"/>
 </partners>
 <variables>
        <variable name="RFDinvoke" messageType="tns:RFDinvoke"/>
   <variable name="RFDmsg" messageType="tns:RFDmsg"/>
        <variable name="RFD_Receipt_Ack" messageType="tns:RFD_Receipt_Ack"/>
        <variable name="Accept" messageType="tns:Accept"/>
        <variable name="DSinvoke" messageType="tns:DSinvoke"/>
        <variable name="DSmsg" messageType="tns:DSmsg"/>
        <variable name="DS_Receipt_Ack" messageType="tns:DS_Receipt_Ack"/>
</variables>

 <correlationSets>
        <correlationSet name="POIdentifier" properties="POIdentifier"/>
        <correlationSet name="RFDIdentifier" properties="RFDIdentifier"/>
</correlationSets>

 <sequence>
   <receive partner="buyer" portType="tns:buyerPT"
                        operation="purchase" variable="RFDinvoke"
                        createInstance="yes" name="ReceivePurchase">
                        <correlations>
                        <correlation set="POIdentifier" initiate="yes"/>
                </correlations>
        </receive>

        <invoke name="invokeRFDoriginator"
                partner="RFDoriginator" portType="tns:RFDoriginatorPT"
                operation="sendRFD" inputVariable="RFDinvoke" outputVariable="RFDmsg">
        </invoke>
        <invoke name="invokeRFDreceiver"
                partner="RFDreceiver" portType="tns:RFDreceiverPT"
                operation="receiveRFD" inputVariable="RFDmsg" outputVariable="RFD_Receipt_Ack">
        </invoke>

        <invoke name="invokeRFD_Accept_Ack"
                partner="RFDreceiver" portType="tns:RFDreceiver"
                operation="sendRFD_Accept" inputVariable="RFDmsg" outputVariable="RFD_Accept_Ack">
        </invoke>
  <invoke name="invokeRFD_Accept_receive"
                partner="RFDoriginator" portType="tns:RFDoriginator"
                operation="receive_Accept" inputVariable="Accept">
        </invoke>
      <invoke name="invokeDS"
                partner="RFDreceiver" portType="tns:RFDreceiver"
                operation="submitDS" inputVariable="DSinvoke" outputVariable="DSmsg">
      </invoke>
   <invoke name="invokeDS_Receipt_Ack"
                partner="RFDoriginator" portType="tns:RFDoriginator"
                operation="receiveDS" inputVariable="DSmsg" outputVariable="DS_Receipt_Ack">
      </invoke>
  </sequence>
</process>
```

The business collaboration or design collaboration patterns can be very complicated because they often involves multiple interactive messages based on the primitive protocols. Take design outsourcing for example. In Figure 8, on the right hand side, it shows a design center, Yamato Design Team, outsource parts of the ThinkPad design to different design partners, i.e to MyASIC for ASIC Chip Design, to MyComputerCorp for Motherboard Design, and to YSAU lab for Mechanical and Electrical Design. On the left-hand side of Figure 8, it shows the design collaboration patterns between Yamato Design Team and MyASIC. The various messages flow between the two partners, starting from a request for design (RFD), followed by request for updates (RFUs), by the acceptance of the RFD, by the submission of design, by the reviewing of the design, by further RFUs, finally concluded with the acceptance of design.

As the sample design collaboration pattern shown Figure 8 demonstrated the flexibility and versatility of the RDF to support various data format required in collaboration message flow and document exchanges.

2.3 HyperChain Manager

As shown in Figure 9, HyperChain Manager is the CxP Engine. This architecture shows that CxP Engine is on the J2EE platform with WebSphere as an example, which is the one we're currently using for eBC POC with the portal deployed on WPS. The portal or dashboard are applications that can access the CxP core engine, enclosed in the big blue box, via the API layer provided by Business Collaboration Manager.

The Collaboration Directory Manager component manages the resources tracked by the CxP engine, such as organizations (partners), users, projects, tasks, etc., and the resources are RDF-based.
The CxP messages are sent and received by the Message Sender and Receiver modules and they are SOAP messages.

The Annotation Manager processes the meta data or annotations created for the documents exchanged via CxP messages. Examples of annotations are file name, file type, version, author name, etc. In addition, annotations can also be used to specify "actions" to be performed on the documents. Examples of such actions may be "review" document, perform "RFTP" (reliable file transfer) and send actions to legacy applications like ERP and PDM, etc..

The annotations in the received messages are forwarded to Action Manager, which is an integration layer to back-end legacy applications as well as components like RFTP, to invoke the proper actions on the documents.

Additionally, As shown in Figure 9B, the Action Manager of the HyperChain Manager is responsible for processing the incoming requests and automatically allocates services for action handling from the distributed service Grid as well as intelligently aggregates data from multiple sources.

The major components of the Action Manger are simply described as follows:

Communicator

To communicate with other collaborators for receiving request and present response
WSDL Interface for monitoring and interaction with other collaborators

Annotation Data Parser

To parse the design HyperChain annotation data

Event Capture

Capture events from design tools such as Cadence, Catia and then pass events to Master controller.

Master Controller

Process the events captured by Event Capture Module (e.g. Communicate with project manager of ID&DE to extract/update the status information)
Sender/receiver verification (e.g. access control list)
Fetch design data, BOM data, other metadata based on the Design HyperChain annotation data
Pass actions defined in the annotation data to Action handler

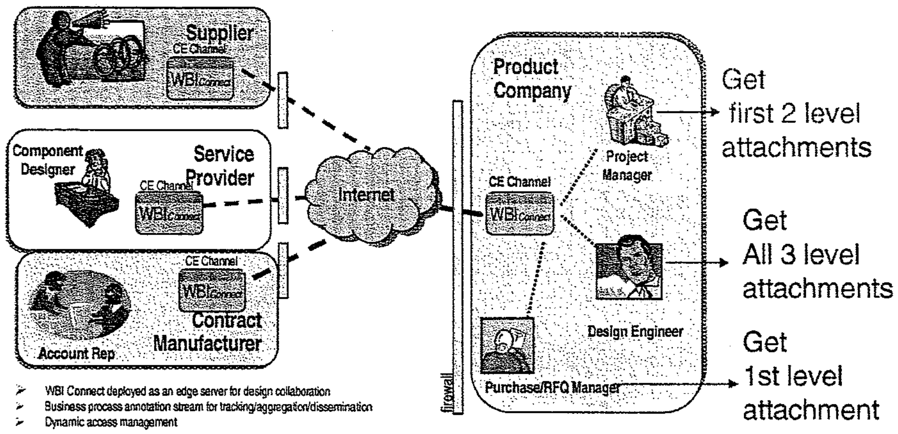
Action Handler

Design Data Annotation, Design Data tracking, and Design Process tracking
Design data aggregation across disparate data sources
Synchronize the collaborative process with design data and metadata
Analyze the results, suggest changes
Annotation data disseminator (access control based, Individual delivery, batch delivery (e.g. broadcast)
Collaboration Status Tracking (Who viewed design documents, what's the status, when viewed, where, and which browser)

Service Grid Powered: On-demand services deployment and delivery model in the design HyperChain engine based on the required action type and real requirements from the Action Handler.

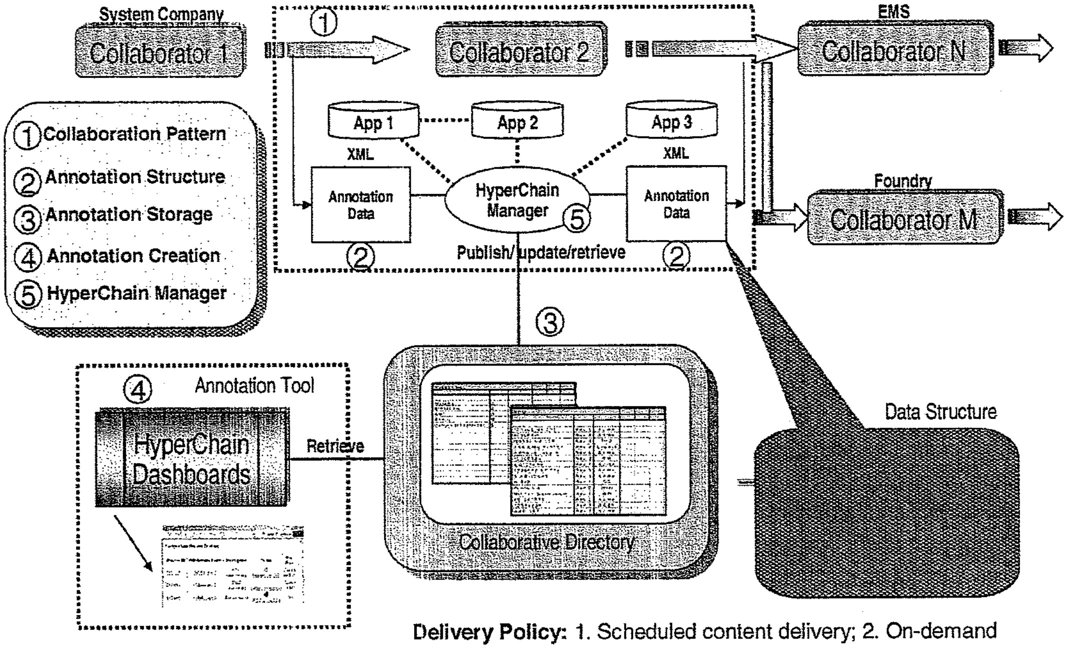
In this disclosure, we present Method and Apparatus of Information HyperChain Management for On-Demand Business Collaboration that includes the following unique ideas and methods:
A. Provide a flexible and uniform annotation representation for information exchange of various non-structured data without requiring pre-defined schemas.
B. Propose a collaboration pattern enabling framework that covers all aspects of a nonstructural and non-deterministic information exchange flow based on the annotation data.
C. Propose an efficient on-demand information manager that dynamically locates resources from the nonstructural relationship graph as well as intelligently aggregates data from multiple sources;
D. Present a trackable information embedding mechanism for visibility control that addresses key issues associated with data exchange in collaborative environment;

While the invention has been described in terms of a single preferred embodiment, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.



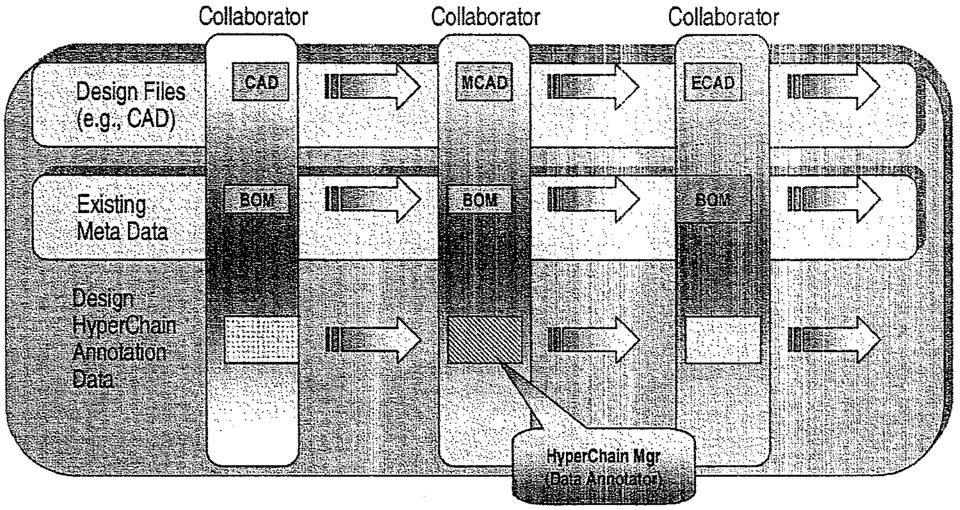
- WBI Connect deployed as an edge server for design collaboration
- Business process annotation stream for tracking/aggregation/dissemination
- Dynamic access management

Figure 1. Example distributed solution architecture



Delivery Policy: 1. Scheduled content delivery; 2. On-demand content delivery (http, ftp; CAD, BOM-LFT); 3. Access control-based content delivery;

Figure 2. HyperChain based On-Demand Information Exchange Model (ODIEM)



*Note - CAD stands for "Computer-Aided Design", and MCAD stands for Mechanical CAD, and ECAD
stands for Electronic CAD.

Figure 3. Design HyperChain Diagram

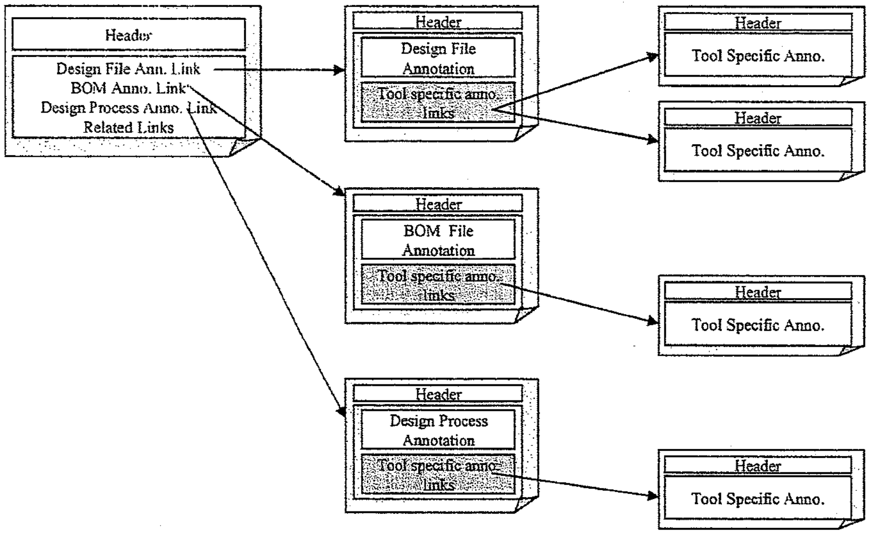


Figure 4. Hierarchical Linkage of the HyperChain Annotation Data Structure

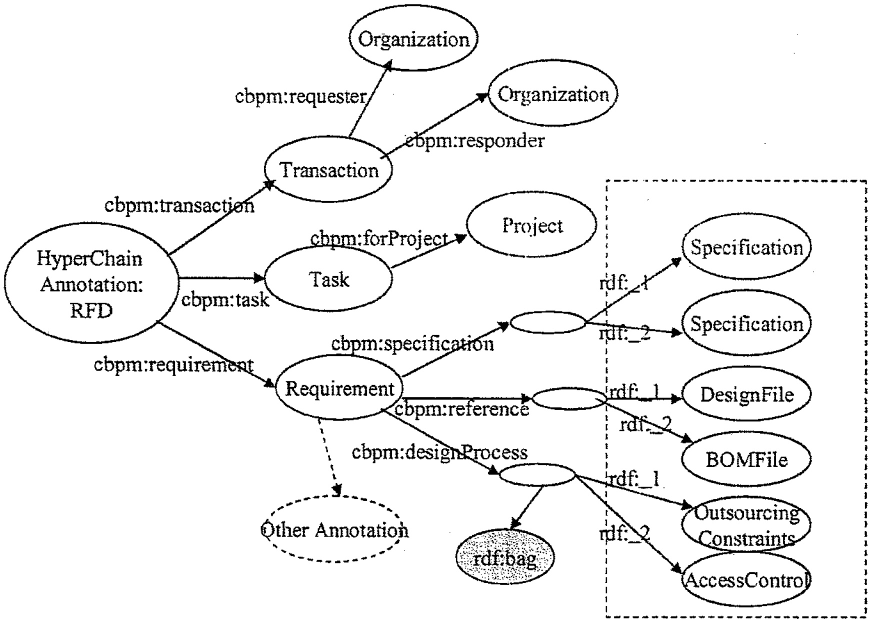


Figure 5. RDF Graph for HyperChain Annotation Data

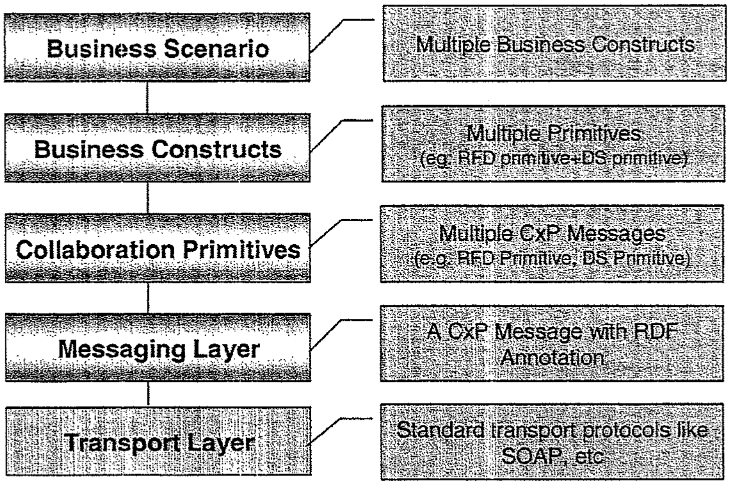


Figure 6. Collaborative Exchange Protocol (CxP) Stack

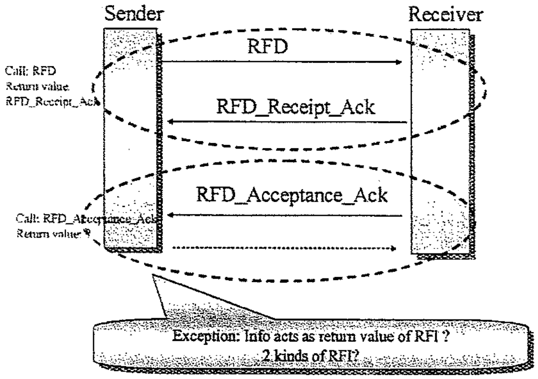
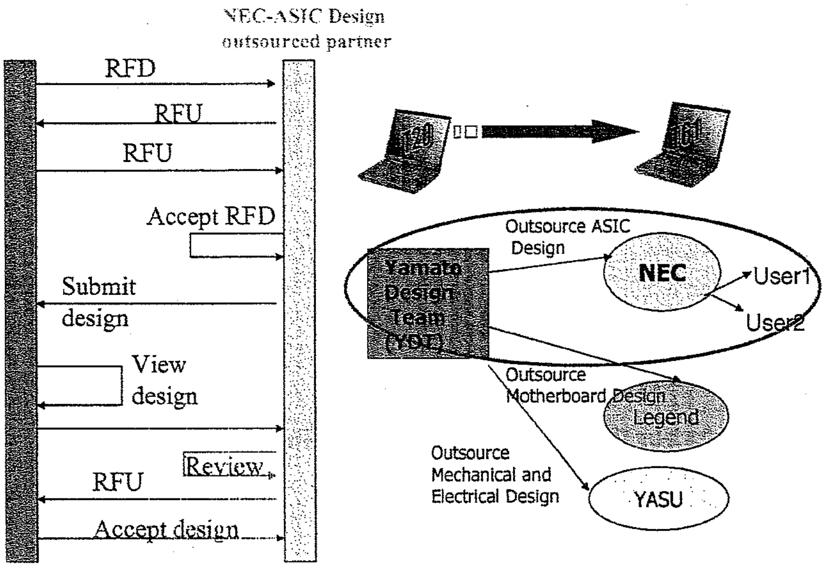


Figure 7. Example Collaboration Pattern (Primitive/Protocol)



Note. RFD – Request for Design, RFU - Request for Update

Figure 8. Example Collaboration Patterns for Outsourcing

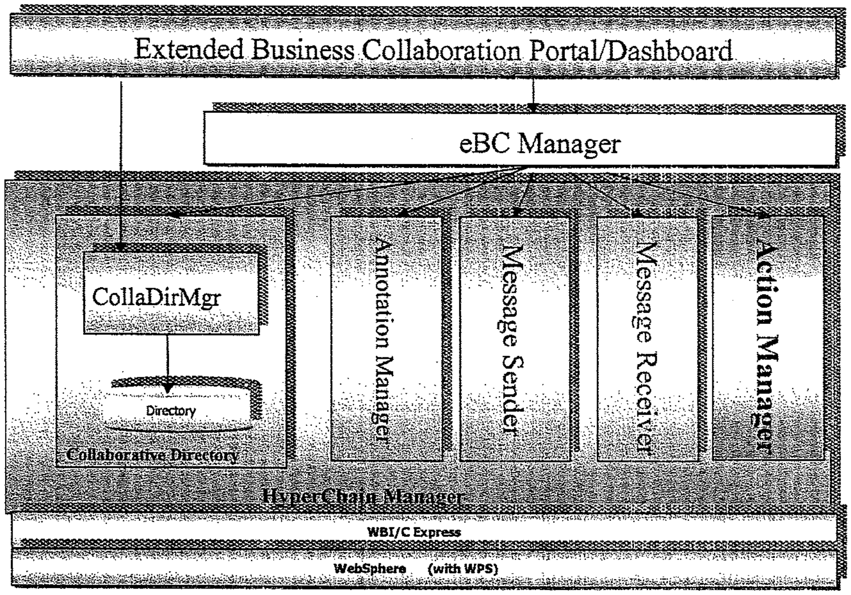


Figure 9A. HyperChain Manager

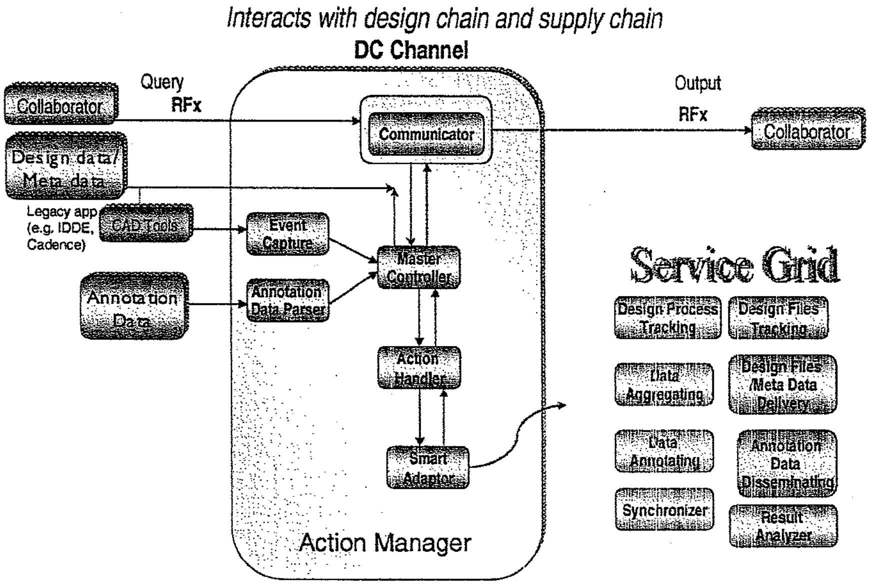
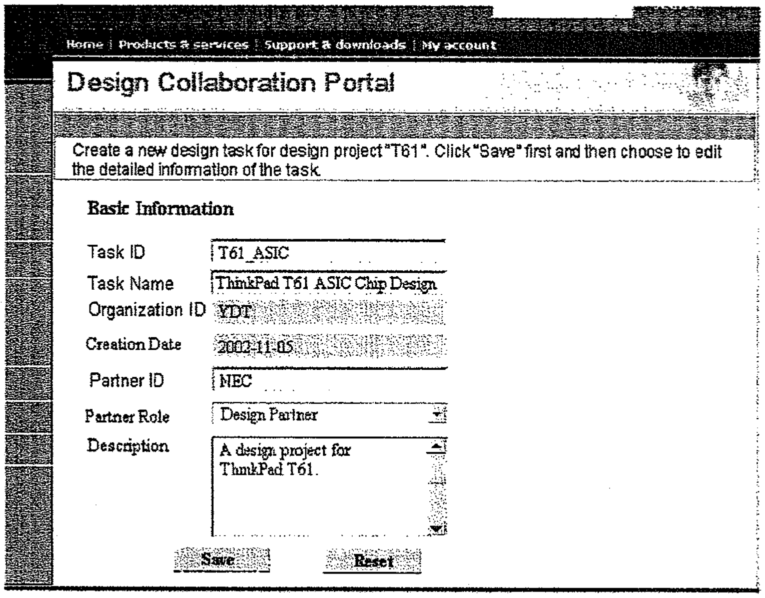


Figure 9B. Action Manager

Figure 9. HyperChain Manager



Home | Products & services | Support & downloads | My account

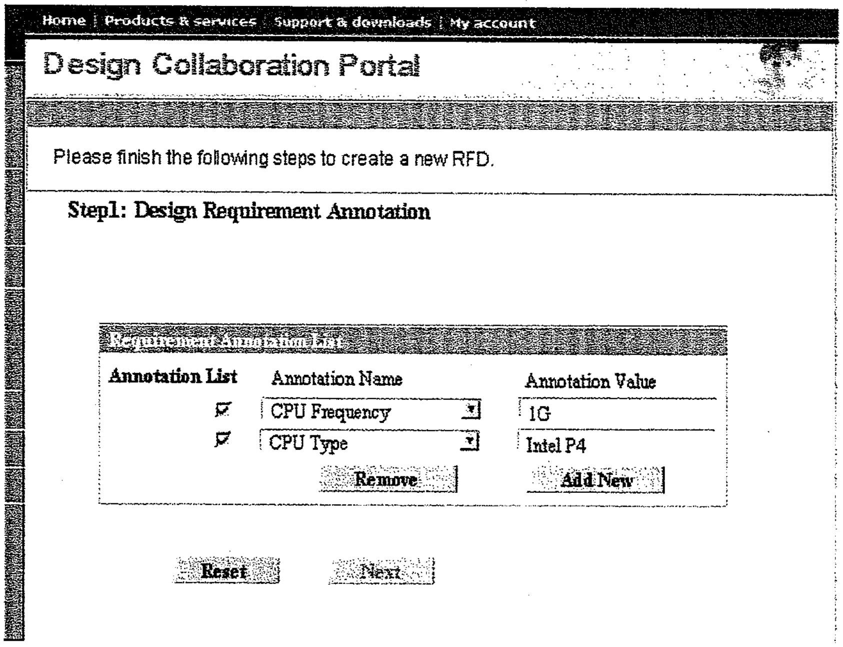
Design Collaboration Portal

Create a new design task for design project "T61". Click "Save" first and then choose to edit the detailed information of the task.

Basic Information

Task ID	T61_ASIC
Task Name	ThinkPad T61 ASIC Chip Design
Organization ID	YDT
Creation Date	2002-11-05
Partner ID	NEC
Partner Role	Design Partner
Description	A design project for ThinkPad T61.

Figure 10. Outsourcing Task Creation Screen


Home | Products & services | Support & downloads | My account

Design Collaboration Portal

Please finish the following steps to create a new RFD.

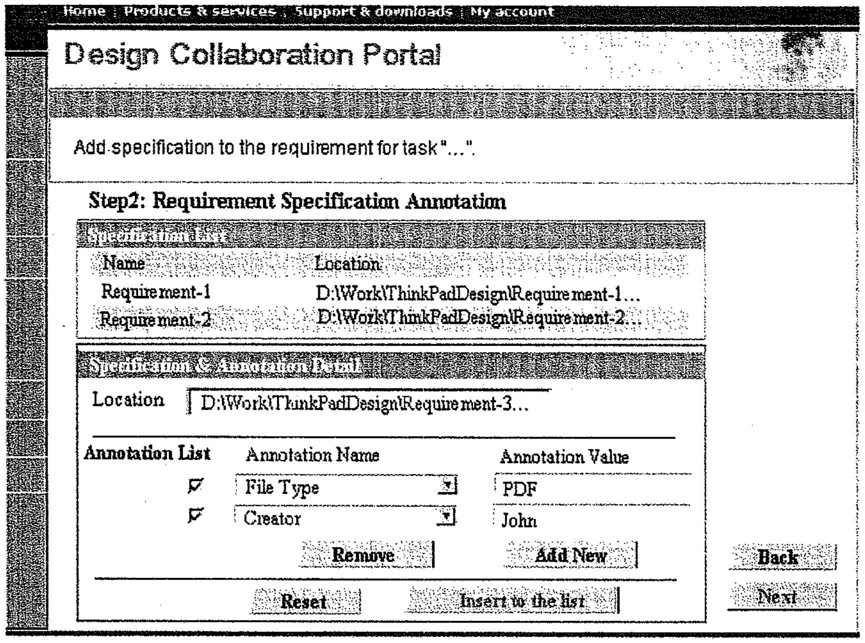
Step1: Design Requirement Annotation

Requirement Annotation List

Annotation List	Annotation Name	Annotation Value
☑	CPU Frequency	1G
☑	CPU Type	Intel P4

Remove
Add New
Reset
Next

Figure 11. Annotation Screen for Design Requirements


Home | Products & services | Support & downloads | My account

Design Collaboration Portal

Add specification to the requirement for task "...".

Step2: Requirement Specification Annotation

Specification List

Name	Location
Requirement-1	D:\Work\ThinkPadDesign\Requirement-1...
Requirement-2	D:\Work\ThinkPadDesign\Requirement-2...

Specification & Annotation Detail

Location D:\Work\ThinkPadDesign\Requirement-3...

Annotation List	Annotation Name	Annotation Value
☑	File Type	PDF
☑	Creator	John

EXHIBIT 2

IBM

Route 100
Somers, NY 10589

VIA OVERNIGHT

July 30, 2003

Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
Attn: William E. Lewis

Re: Patent Application for "Method and Apparatus of Information HyperChain Management for On-Demand Business Collaboration"
Disclosure No.: SOM8-2003-0001
Docket No.: SOM920030006US1

Dear Bill,

Enclosed are copies of the above-referenced disclosure and a written embodiment thereof, including drawings. We will send you a soft copy of the embodiment in an e-mail. Please refer to the docket number in all correspondence.

Please prepare and file a patent application no later than September 30, 2003. The inventor you should work with is John Sayah, who can be reached at 1-914-784-7040.

Best regards,

David M. Shofi

DMS/lbd
Enclosures
cc: John Sayah w/o enclosures

EXHIBIT 3

William E. Lewis

From: "William E. Lewis" <wel@rml-law.com>
To: <tian@us.ibm.com>
Cc: <zhanglj@us.ibm.com>; <john_sayah@us.ibm.com>
Sent: Thursday, September 11, 2003 10:00 AM
Attach: SOM920030006US1.ZIP
Subject: SOM920030006US1 (Hyperchain)

Tian:
As we discussed, please find attached a first draft of the hyperchain application. Since the drawings are in a .tif format (which I could not individually password-protect) and the application itself is in WordPro format, I zipped the two files and protected the zipped file with the password we agreed upon yesterday.

Thank you for your patience and assistance.
Regards,
Bill

William E. Lewis
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, New York 11560
Telephone: 516-759-2946
Fax: 516-759-9512
E-mail: wel@rml-law.com

CONFIDENTIALITY NOTICE
This message may contain legally privileged and/or confidential information intended only for the recipient(s) identified above. If you are not an intended recipient, you are hereby notified that any dissemination, distribution, copying or other use of any portion of this message, including any attachments, is strictly prohibited. If you have received this message in error, please immediately notify the sender by telephone or reply e-mail and delete this message. Thank you.

EXHIBIT 4

William E. Lewis

From: William E. Lewis [wel@rml-law.com]
Sent: Thursday, September 18, 2003 4:18 PM
To: Liang-Jie Zhang
Cc: John Sayah; Tian Chao
Subject: Re: SOM920030006US1 (Hyperchain) - updates

To all:
Please find attached a revised draft of the application with a draft claims set.
The password is the same as before. Please get back to me as early as possible tomorrow with any final comments. Once we finalize the claims, I have to run them by Dave Shofi to get his approval to file. If all goes well, we should be able to file tomorrow afternoon. Please confirm receipt of the attachment.
Regards,
Bill

William E. Lewis
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, New York 11560
Telephone: 516-759-2946
Fax: 516-759-9512
E-mail: wel@rml-law.com

CONFIDENTIALITY NOTICE
This message may contain legally privileged and/or confidential information intended only for the recipient(s) identified above. If you are not an intended recipient, you are hereby notified that any dissemination, distribution, copying or other use of any portion of this message, including any attachments, is strictly prohibited. If you have received this message in error, please immediately notify the sender by telephone or reply e-mail and delete this message. Thank you.

> ----- Original Message -----
> **From:** Liang-Jie Zhang
> **To:** William E. Lewis
> **Cc:** John Sayah ; Tian Chao
> **Sent:** Wednesday, September 17, 2003 2:41 PM
> **Subject:** Re: SOM920030006US1 (Hyperchain) - updates
>
> Hi Bill,
>
> Thanks a lot. We will send you feedback according to your schedule.
>
> Best Regards,
> LJ

William E. Lewis

From: William E. Lewis [wel@rml-law.com]
Sent: Wednesday, September 17, 2003 2:25 PM
To: Liang-Jie Zhang
Cc: John Sayah; tian@us.ibm.com
Subject: Re: SOM920030006US1 (Hyperchain) - updates

LJ:
I am in the process of revising the application and putting together the claim set. I hope to have it to you by tomorrow. If you can review the revised draft and return any final comments by early Friday morning, we should be able to get the case filed by Friday afternoon (subject to Dave Shofi's approval).
Regards,
Bill

William E. Lewis
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, New York 11560
Telephone: 516-759-2946
Fax: 516-759-9512
E-mail: wel@rml-law.com

CONFIDENTIALITY NOTICE
This message may contain legally privileged and/or confidential information intended only for the recipient(s) identified above. If you are not an intended recipient, you are hereby notified that any dissemination, distribution, copying or other use of any portion of this message, including any attachments, is strictly prohibited. If you have received this message in error, please immediately notify the sender by telephone or reply e-mail and delete this message. Thank you.

> ----- Original Message -----
> **From:** Liang-Jie Zhang
> **To:** William E. Lewis
> **Cc:** John Sayah
> **Sent:** Wednesday, September 17, 2003 12:13 PM
> **Subject:** Re: SOM920030006US1 (Hyperchain) - updates
>
> Dear Bill,
>
> Thank you very much for your help. It looks in a good shape. If you need to discuss the claims, John and myself will be very happy to join you in a conference call.
>
> Best Regards,
> LJ
>
> -----------------------------------
> Liang-Jie (LJ) Zhang (Ph.D.), Research Staff Member
> Web Services/Grid Solutions for Business Process Integration
> e-Business Solutions & Autonomic Computing Dept.
> IBM T.J. Watson Research Center

William E. Lewis

From: William E. Lewis [wel@rml-law.com]
Sent: Tuesday, September 16, 2003 10:56 AM
To: Liang-Jie Zhang
Cc: tian@us.ibm.com; john_sayah@us.ibm.com
Subject: Re: SOM920030006US1 (Hyperchain) - updates

Hi Tian/LJ:
Thank you for the comments : I will incorporate them in a revised draft and I will have the revised draft to you as soon as possible.

Regards,
Bill

William E. Lewis
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, New York 11560
Telephone: 516-759-2946
Fax: 516-759-9512
E-mail: wel@rml-law.com

CONFIDENTIALITY NOTICE
This message may contain legally privileged and/or confidential information intended only for the recipient(s) identified above. If you are not an intended recipient, you are hereby notified that any dissemination, distribution, copying or other use of any portion of this message, including any attachments, is strictly prohibited. If you have received this message in error, please immediately notify the sender by telephone or reply e-mail and delete this message. Thank you.

> ----- Original Message -----
> **From:** Liang-Jie Zhang
> **To:** William E. Lewis ; Tian Chao
> **Cc:** John Sayah
> **Sent:** Monday, September 15, 2003 2:08 PM
> **Subject:** Re: SOM920030006US1 (Hyperchain) - updates
>
> Hi Bill and Tian,
>
> . Attached please find the update. Thanks.
>
> Best Regards,
> LJ